

CODE FOR COMPUTER BRAILLE NOTATION

Pamphlet File
Research Library
APH

Compiled Under the Authority of the
BRAILLE AUTHORITY OF
NORTH AMERICA

Department of
Educational Research

Adopted May 1, 1987

AMERICAN PRINTING HOUSE FOR THE BLIND
Louisville, Kentucky 40206-0085
1987

7-24430

CODE FOR COMPUTER BRAILLE NOTATION

Pamphlet File
Research Library
APH

Department of
Educational Research

Compiled Under the Authority of the
BRAILLE AUTHORITY OF
NORTH AMERICA

Adopted May 1, 1987

AMERICAN PRINTING HOUSE FOR THE BLIND
Louisville, Kentucky 40206-0085
1987

BRaille AUTHORITY OF NORTH AMERICA

**Charlotte Begley
Darleen Bogart
Floyd R. Cargill
Betty Epstein
Dolores Ferrara-Godzieba
Marc Maurer
Ralph E. McCracken
Miriam Pace
Dr. Susan Spungin
Dr. Robert Winn**

FOREWORD

As our society's dependence on computing technology grows, computers become increasingly important to the blind and visually impaired. Those who read braille in order to learn about computers and to communicate with them have an urgent need for a braille computer code that is unambiguous, and that makes possible the faithful transcription into braille of computer-related text in print. In 1972, BANA recognized this need by adopting The Provisional Braille Code for Computer Notation. It was a preliminary code that left many problems unsolved, but its adoption allowed braille readers and braille transcribers to accumulate a body of experience on which to base a more efficient code. The code presented and explicated in this volume has been made possible by that experience.

The Computer Braille Code utilizes the sixty-four combinations of dots that can be formed in the braille cell, and assigns to these dot patterns the same meanings assigned to them by North American manufacturers of computer braille equipment. However, an effective braille code must be capable of representing the 128 characters in the ASCII code, and must provide a few additional symbols that transcribers need in order to achieve faithful braille transcriptions. To meet these requirements, two meanings must be assigned to many of the sixty-four dot patterns, and a way of indicating the meaning that is intended must be provided. In the Computer Braille Code, the differentiation between the two meanings assigned to a dot pattern is accomplished by means of a prefix consisting of dots 4-5-6.

The Computer Braille Code Committee has studied a large and diverse sample of current computer materials, and has worked diligently to make the code for computer braille notation a realistic code, capable of unambiguous representation of current computer notation, but flexible enough to respond to changing and expanding needs. Feedback from braille readers, transcribers, and publishers has been solicited, and is reflected in this edition.

BANA and this committee recognize that the code does not address important issues, such as the transcription of flow charts and graphics. However, we are pleased to publish it in its present form, and trust that a future revision will deal with such issues. We welcome suggestions and comments from the field.

BANA wishes to thank Dr. T. V. Cranmer (Chairman of the Computer Braille Code Committee), and Dr. Emerson Foulke, Mrs. Priscilla Harris, Miss Donna Pastore, Dr. Sandra Ruconich, Dr. Lawrence Scadden, and Mr. Joseph Sullivan (committee members), for their work.

CONTENTS

BANA MEMBERS	ii
FOREWORD	iii
COMPUTER BRAILLE CODE	
I - Guidelines for Identifying Computer Notation	1
A. A Note	1
Example 1-1 Note in Text	1
Example 1-2 Dot Matrix Print	2
B. Program Lines	2
Example 1-3 Short Program Lines Identified by Numbers	2
Example 1-4 Systematic Indentation of Program Lines	2
C. Comments	3
Example 1-5 Explanatory Comments	3
Example 1-6 Explanatory Comments	3
D. Words	3
Example 1-7 Word Enclosed within Angle Brackets	3
E. Horizontal Signs	3
Example 1-8 Horizontal Sign of Comparison	3
Example 1-9 Horizontal Sign of Comparsion	4
F. Superimposed Symbols	4
Example 1-10 Superimposed Symbol	4
G. Words and/or Symbols	4
Example 1-11 Words Out of Context	4
II - Computer Braille Code Symbols and Usage	4
A. The Basic of the Computer Braille Code	4
B. Use of Braille Symbols	5
C. The Computer Braille Code - A Separate Code	5
Example 2-1 Technical-Appearing Text (Literary Code)	5
Example 2-2 Text with Embedded Computer Notation	6
Example 2-3 Computer Program	6
Example 2-4 Nonuse of to, into, and by Preceding Computer Braille Code Indicator	6
III - Format	6
A. Computer Braille Code Transcriptions	6
B. Transcriber's Note	6
C. Symbols	7
D. Blank Lines	7
Example 3-1 Computer Notation Set Off from Body of Text	7
4.4 For 360/91 Users Only	7
E. Entire Computer Program	8
Example 3-2 Program with Identifying Label	8

F. Computer Notation within Text	9
Example 3-3 Embedded Computer Notation	9
Example 3-4 Embedded Computer Notation with Runover	9
Example 3-5 Embedded Computer Notation on New Line	10
IV - Upper and Lower Case Designation	10
A. Computer Notation / Lower Case	10
Example 4-1 Single BASIC Program Transcribed without Case Indicators	10
B. Shift Indicator	11
C. Caps Lock Indicator	11
Example 4-2 Consecutive and Single Upper Case Letters	11
D. Caps Release Indicator	11
Example 4-3 Caps Lock and Caps Release Indicators	11
V - Representation of Control Characters	11
Example 5-1 "Carriage Return" Representation	12
Example 5-2 "Carriage Return" Representation	12
VI - Continuation Indicator	12
A. Continuation Indicator	12
Example 6-1 Continuation Indicator	12
B. Runovers of Computer Notation	12
C. Continuation Indicator	13
Example 6-2 Continuation Indicator within Embedded Computer Notation	13
VII - Indented Material	13
A. Levels of Indentation	13
B. Runovers	13
Example 7-1 Indented Material	13
Example 7-2 Indented Material	14
VIII - Isolated Lower-Cell Signs	14
Example 8 Isolated Lower-Cell Signs	15
IX - Countable Spaces	15
A. Countable Spaces Indicator	15
Example 9-1 Countable Spaces Indicator	15
B. Countable Spaces Indicator	15
Example 9-2 Countable Spaces with Continuation Indicator	15
Example 9-3 Countable Spaces Beginning on Runover Line	16
X - Emphasis Indicators	16
A. Emphasis Indicators	16
Example 10-1 Emphasis Indicators	16
Example 10-2 Emphasis Indicators	17
B. Type Style	17
Example 10-3 Emphasis Indicators with More Than One Type Style	17

XI - Computer Braille Code Indicators	18
A. Computer Braille Code Indicators	18
B. Caps Lock Indicator	18
Example 11-1 Computer Braille Code Indicators	18
Example 11-2 Avoiding Frequent Switches Between Computer Braille Code and Primary Code	18
Example 11-3 Use of Computer Braille Code Indicators to Clarify Ambiguities	18
XII - Nemeth Code Indicators	19
A. Nemeth Code	19
Example 12-1 Nemeth Code Indicators	19
Example 12-2 Nemeth Code Indicators with Horizontal Fraction Line	20
B. Nemeth Code Indicators	20
Example 12-3 Nemeth Code Indicators within Computer Braille Code	20
XIII - Shape Indicators	20
A. Shape Indicators	21
Example 13-1 Shape Indicators	21
B. Computer Keyboard	21
Example 13-2 Depicted Key	21
C. Transcribers of Nemeth Code	21
Example 13-3 Use of Nemeth Code within Shape Indicators	21
D. A Transcriber's Note	22
XIV - Transcriber's Option Symbols	22
A. Transcriber's Option Symbol	22
Example 14-1 Transcriber's Option Symbol Representing Non-ASCII Character (b)	22
B. Transcriber's Substitution	22
Example 14-2 Transcriber's Substitution (ASCII Tilde Becomes EBCDIC "Not" Sign)	22
XV - Half-Line Shifts	23
A. Half-Line Shifts Up	23
B. Half-Line Shifts Down	23
Example 15-1 Half-Line Shifts	23
Example 15-2 Half-Line Shifts Terminated by A Space	24
C. Reference Indicators	24
Example 15-3 Reference Indicator (English Braille)	24
Example 15-4 Reference Indicator (Nemeth Code)	24
XVI - Termination Indicator	25
Example 16 Termination Indicator	25
COMPUTER BRAILLE CODE SYMBOLS	
ASCII ORDER	26
ADDITIONAL COMPUTER BRAILLE CODE SYMBOLS	31
BRAILLE ORDER	33

Computer Braille Code

I — Guidelines for Identifying Computer Notation

The Computer Braille Code is entered when precise representation is paramount. Thus, material such as programs, program lines, computer commands, and filenames should be transcribed in Computer Braille Code. The following guidelines may be helpful in the identification of computer notation. Because languages and situations the transcriber may encounter will doubtless vary widely, these guidelines are offered merely as suggestions and not as all-inclusive, binding rules.

A. A note at the beginning of a work may describe how computer notation is represented. Even if no such note exists, stylistic practices (e.g., type styles different from surrounding text—dot matrix print, for instance) may indicate the presence of computer notation.

Example 1-1: Note in Text

Your console displays both what you type and the computer response. This book uses the following convention to differentiate between output from the computer and data typed at the keyboard: All operator input is underlined, while the computer-generated output is not.

Here is an example:

```
A > DIR*.COM < cr >
A:      FORMAT.COM:MBASIC.COM:PIP.COM:ED.COM
```

Example 1-2: Dot Matrix Print

DOS READY

B. Program lines are generally relatively short and sometimes begin with numbers. Often these lines are indented in a systematic way which distinguishes them from mathematics or regular text.

Example 1-3: Short Program Lines Identified by Numbers

```
10 PRINT "HOW OLD ARE YOU?"
```

20 INPUT A

30 PRINT "THANK YOU!"

40 PRINT "YOU CAN RETIRE"

50 PRINT "IN" ; 65-A ; "YEARS."

60 END

The image displays a 6x10 grid of 60 small clusters of black dots on a light gray background. Each cluster represents a different arrangement of 10 dots, totaling 60 unique permutations of the 10-dot set. The dots are arranged in various patterns, including linear, rectangular, and more complex geometric shapes, illustrating the concept of permutations in a visual and systematic way.

Example 1-4: Systematic Indentation of Program Lines

for I := 1 to NoOfLines do

 $\overline{\text{begin}}$

Readln (Line);

if Length (Line) < Limit then

$$\text{ShortLines} := \text{ShortLines} + 1$$

else

$$\overline{\text{LongLines}} := \text{LongLines} + 1$$

end;

C. Comments sometimes explain what is happening in a program. These comments are frequently distinguished from surrounding material by characteristic symbols or words [e.g., /* ... */ , (* ... *) , REM, OUTPUT, etc.].

Example 1-5: Explanatory Comments

- `/* This is a comment in C */`
- `(* This is a comment in Pascal *)`
- `{ So is this }`
- `110 REM THIS IS A COMMENT IN BASIC`

Example 1-6: Explanatory Comments

{ Longest possible representation }

D. Words may be enclosed within quotation marks or angle brackets.

Example 1-7: Word Enclosed within Angle Brackets

< FILE >

E. Horizontal signs of comparison (e.g., $\leq$, $=$, $:=$) may be present.

Example 1-8: Horizontal Sign of Comparison

```
if (inword == 0)
```

Example 1-9: Horizontal Sign of Comparison

$$(ev \sim= evNull) \Rightarrow$$

F. Superimposed symbols (e.g., a circle with a slash inside) may appear.

Example 1-10: Superimposed Symbol

To OVERLAY one character on top of another for the purpose of having two or more characters at the same character position at print time, e.g., $\neq$:

Note: The shape indicators enclose 2h1v to describe the shape consisting of two horizontal lines and one vertical line.

G. Words and/or symbols may not make sense in the context of the primary code being used.

Example 1-11: Words Out of Context

10 LPRINT “HELLO”

II — Computer Braille Code Symbols and Usage

A. The basis of the Computer Braille Code is ASCII (American Standard Code for Information Interchange), a code used by virtually all computer-related braille devices distributed in North America. Printable symbols from other codes (e.g., EBCDIC and BAUDOT) can also be represented using Computer Braille Code symbology. The Computer Braille Code

makes communication with braille devices possible while simultaneously providing for the precise transcription of braille computer-related materials. A table of Computer Braille Code symbols appears at the end of this code.

B. Although in certain instances many braille devices use one braille symbol to represent two different print characters (e.g., upper case A and lower case a are both brailled as dot 1), the Computer Braille Code avoids this ambiguity. The same braille symbol is used for both characters, but the symbol representing the second print character is preceded by a differentiating prefix, dots 4-5-6. Braille symbols used in this dual manner include single upper case (A) and lower case (a) letters, at sign (@) and grave accent (`), up arrow/circumflex ($\uparrow$ $\wedge$) and tilde (~), left bracket ([) and left brace ({), right bracket (]) and right brace (}), and backslash (\) and vertical bar (|). The underscore symbol (__) has a separate identity.

C. The Computer Braille Code is a separate code which may be used in conjunction with the primary code in which the text is transcribed—whether that code be English braille, textbook format, Nemeth, or braille music. Material such as numbers, abbreviations, acronyms, contractions, and punctuation marks, should be transcribed in this primary code unless there is good reason to enter Computer Braille Code. Thus, technical-appearing abbreviations and acronyms such as “IBM” and “ASCII” occurring within an English braille transcription should be transcribed in English braille. Computer Braille Code is entered when precise representation is paramount; for this reason, Computer Braille Code material is transcribed character for character (i.e., no contractions are used), and the contractions to, into, and by may not be used before any Computer Braille Code indicator. Computer Braille Code numbers are transcribed in the lower two-thirds of the cell and are not preceded by number signs. Programs and other computer notation set off from the body of the text and computer input in which every letter must be typed precisely in the manner shown are examples of material which should be transcribed in Computer Braille Code. When in doubt, Computer Braille Code should be used. Likewise, the transcriber must also judge or seek advice as to the significance of vertical and horizontal spacing within the text. When in doubt, assume spacing is significant.

Example 2-1: Technical-Appearing Text (Literary Code)

There were many complaints about the size of BCDIC, and IBM introduced Extended BCDIC with System/360.

Example 2-1: Text with Embedded Computer Notation (English Braille and Computer Braille Codes)

Example 2-2: Text with Embedded Computer Notation (English Braille and Computer Braille Codes)

The test criterion compares j to the term value (2) 3000 ($\neq$ tests for inequality).

Example 2-3: Computer Program (Computer Braille Code)

```
struct nlist *np, *lookup();
char *strsave(), *alloc();
int hashval;
```

Example 2-4: Nonuse of to, into, and by Preceding Computer Braille Code Indicator

Example 2-4: Nonuse of to, into, and by Preceding Computer Braille Code Indicator

This number divided by OFF_{16} is the fraction of the symbol table used.

Note: This example is in hexadecimal notation. In this notation, the numbers are the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and the letters A, B, C, D, E, F. Therefore, the example shows the number OFF not the word OFF.

III — Format

A. Computer Braille Code transcriptions should use the longest possible braille line allowed by production and duplication considerations. Current line length is typically 40 or 41 cells.

B. A transcriber's note at the beginning of each volume of a book should state the year of adoption of the Computer Braille Code, as well as the primary code in which the book is transcribed. If a book requires more than two codes, all codes employed should be listed.

C. Symbols which the transcriber has devised or those assigned special usage should be explained or drawn on a "Special Symbols" page. These symbols should be transcribed in accordance with the rules of Textbook Format Code. Neither standard Computer Braille Code symbols nor symbols of other adopted codes which are used in accordance with the rules of the applicable code(s) should be included on the "Special Symbols" page.

D. A blank line should precede and follow programs and computer notation set off from the body of the text. Transcription of such material should begin in cell 1 of the appropriate line, with runovers beginning in cell 2 of the following line(s). (See Sections VI, VII, and IX for more information about runovers.)

Example 3-1: Computer Notation Set Off from Body of Text

The following JCL may be used when the above steps are complete, or as a third step of the above, to generate the compiler:

```
//LISP JOB  
//STEP3 EXEC PGM=LISP  
//STEPLIB DD DSNAME=LISP,VOLUME=SER=zzzzzz,UNIT=www,DISP=OLD  
//LISPOUT DD SYSOUT=A
```

4.4 For 360/91 Users Only

Because of hardware differences on the IBM 360/91, the following update cards must be included during the assembly of . . .

```
      . . . . .  
      . . . . .  
      . . . . .
```

```
      . . . . .  
      . . . . .  
      . . . . .  
      . . . . .  
      . . . . .  
      . . . . .
```

```
      . . . . .  
      . . . . .  
      . . . . .  
      . . . . .  
      . . . . .
```

E. Whenever possible, an entire computer program should be transcribed on a single braille page. When a program is too long to fit on a single page, any identifying label should be centered on line 25 of each page to which it applies. A program line cannot be divided between pages. Text following a program requiring more than one braille page should begin on a new page, with the transition to the new page replacing the blank line which ordinarily follows a program.

Example 3-2: Program with Identifying Label

```

Program 3-5
100 CLS
110 A$ = "FROG"
120 INPUT "JUMP = "; J
130 P = 3 : C = 0
140 CLS : PRINT A$;
150 FOR X = 1 TO 500 : NEXT X
160 IF P >= 60 GOTO 210
170 PRINT TAB(POS(2)+J) A$;
180 FOR X = 1 TO 200 : NEXT X
190 P = P + 4 + J : C = C + 1
200 GOTO 160
210 FOR X = 1 TO 300 : NEXT X
220 CLS : PRINT "THE FROG JUMPED OFF THE SCREEN IN";C;"JUMPS"
230 PRINT "HIS LEAP WAS";J;"SPACES"

```

A large grid of 100 rows and 100 columns of dots. Each dot in the grid is replaced by a small cluster of dots, creating a complex, fractal-like pattern. The clusters vary in size and shape, but they all seem to be composed of a small number of dots (usually 2 to 5) arranged in a specific way. The overall effect is a dense, textured field of dots that changes as the viewer's perspective shifts.

F. A blank line need not precede or follow computer notation within text (hereafter referred to as embedded computer notation). However, embedded computer notation should be transcribed in Computer Braille Code. If embedded computer notation will not fit entirely on the current line, this notation should begin on a new line; runovers should be written at the margin. Text which follows embedded computer notation may resume on the current line. The begin Computer Braille Code indicator must begin embedded computer notation except when the first symbol of such notation is the caps lock indicator, the begin emphasis indicator, the begin Nemeth code indicator, or the begin shape indicator. A termination indicator must always be used within embedded computer notation to signal a return to primary code and/or to cancel the effect of the most recent instance in which the termination indicator applies. (For discussions of the caps lock, emphasis, Nemeth code, shape, and termination indicators, see Sections IV, X, XII, XIII, and XVI, respectively.)

Example 3-3: Embedded Computer Notation

To specify all files named ACCOUNTS, regardless of extension, type the following:

Example 3-4: Embedded Computer Notation with Runover

The macro command will change all occurrences of ROM to RAM. If used on the example in the “J” commands, the resulting line will read: WHEN IN RAM DO AS THE RAMANS DO, AND BE RAMANTIC...

Note: The words “ROM” and “RAM” are part of a discussion of text editing and are not used as acronyms.

Example 3-5: Embedded Computer Notation on New Line

COBOL

Common Business Oriented Language instructs the computer in English-like sentences; "ADD SALES-AMT TO TOTAL-AREA.". COBOL programs are easy to read, but require a lot of space.

Common Business Oriented Language instructs the computer in English-like sentences; "ADD SALES-AMT TO TOTAL-AREA.". COBOL programs are easy to read, but require a lot of space.

IV — Upper and Lower Case Designation

A. Unless otherwise indicated, the Computer Braille Code assumes that letters and words contained in computer notation are lower case. When computer notation is predominantly or exclusively upper case (e.g., a BASIC text), the transcriber may elect to use single-cell letters to indicate upper case and letters preceded by dots 4-5-6, the shift indicator, to designate lower case. If computer notation throughout a text is exclusively upper case, a transcriber's note to this effect should appear on the "Transcriber's Notes" page. (Examples 1-2, 1-3, 1-5d, 1-7, 1-11, 3-2, 5-1, 6-2, 8, 9-1, and 14-2 would include such a note.) If computer notation throughout a text is not exclusively upper case, the transcriber's note should immediately precede the material to which it applies.

Example 4-1: Single BASIC Program Transcribed without Case Indicators

10 INPUT "YOUR FIRST NAME";A\$
20 INPUT "YOUR MIDDLE NAME";B\$
30 INPUT "YOUR LAST NAME";C\$

10 INPUT "YOUR FIRST NAME";A\$
20 INPUT "YOUR MIDDLE NAME";B\$
30 INPUT "YOUR LAST NAME";C\$

Shift indicator: Dots 4-5-6 ∴

B. The shift indicator precedes a single upper case letter appearing in embedded computer notation and/or computer notation set off from the body of the text which is otherwise in lower case. Two or more upper case letters occurring in predominantly lower case text—whether these letters appear consecutively or are separated by numbers, punctuation, etc.—must be preceded by the “caps lock” indicator (see Part C.).

Caps lock indicator: Dots 4-5-6, 3-4-5 ⋮ ⋮

C. The caps lock indicator is used in instances in which embedded computer notation and/or computer notation set off from the body of the text appears in a combination of upper and lower case. The caps lock indicator remains in effect until cancelled by the caps release indicator or the next significant space, which ever comes first. Neither cells occupied by the continuation indicator, spaces following the continuation indicator, nor the space showing indentation of a runover are considered significant spaces.

Example 4-2: Consecutive and Single Upper Case Letters

UNDEFINED; \$(Implementation detail)

Caps release indicator: Dots 4-5-6, 1-2-6 :: ::

D. The caps release indicator cancels the effect of the caps lock indicator.

Example 4-3: Caps Lock and Caps Release Indicators

VFUN PSTmsgToVec(ipcMsg

V — Representation of Control Characters

Control characters, whether in ASCII or other computer codes, are not actually printed by computers and are therefore symbolized in print in a variety of ways. For example, the carriage return function may be represented by “RETURN,” “ENTER,” “CR,” a downward-curve arrow, a depicted key, or in various other ways. Thus, the braille transcription should follow the print symbology literally, in accordance with the other rules of this code.

Example 5-1: “Carriage Return” Representation

```
) CREATE/I DSWITCH.CLI ↓
```

Note: The downward curved arrow is used as a carriage return. Therefore, in this example the letters cr are used within the shape indicators.

Example 5-2: “Carriage Return” Representation

RETURN

A 3x12 grid of dots forming a stylized representation of the word "BIOETHICS". The dots are arranged in three rows and twelve columns. The first row contains 12 dots, the second row contains 12 dots, and the third row contains 12 dots. The dots are arranged in a way that they form the letters of the word "BIOETHICS" when read from left to right.

VI — Continuation Indicator

Continuation indicator: Dots 4-5-6, 1-2-3-4-6 :: ::

A. The continuation indicator is a line division sign. It is used in instances in which the linear arrangement of the print is significant and the braille line is not long enough to finish the print line. Line division may occur within embedded computer notation as well as within computer notation set off from the body of the text and may generally occur at convenient points (i.e., all available spaces on the braille line need not be filled). If no spaces are present at the point of line division, the continuation indicator should immediately follow the last cell of the braille line (i.e., no spaces should intervene). The continuation indicator may not be used between the cells of a multi-cell sign (e.g., between the first and second cells of the “left brace” symbol). When used to divide countable spaces indicators, the continuation indicator must be preceded by two full cells (dots 1-2-3-4-5-6, 1-2-3-4-5-6). The continuation indicator is not considered a countable space. (See Section IX for a discussion of the countable spaces indicator and its use.)

Example 6-1: Continuation Indicator

```
OFUN PSTipcqPut(seid sender,receiver;evType ev;BOOLEAN pseudoInt;
```

B. The remainder of the original print line is continued on the next braille line(s). Runovers of computer notation set off from the body of the text

begin in cell 2. Runovers of embedded computer notation begin at the margin in order to aid the scanning braille reader in distinguishing program from narrative.

C. A continuation indicator should be used when the entire print line will not fit on the braille line(s), even if the line division occurs following a complete word.

Example 6-2: Continuation Indicator within Embedded Computer Notation

12. Complete the following program, which is similar to that of frame 11. Use string variables for substrings to produce this display from `$Z = "THE END OF A STRING IS ON THE RIGHT."`

A 5x10 grid of 50 small, stylized human figures in various poses, representing a population.

VII — Indented Material

A. When successive levels of indentation occur in print, braille lines should also be indented. Each braille level of indentation should begin two spaces to the right of the previous level, regardless of the print spacing used. Thus, successive levels of braille indentation begin in cells 1, 3, 5, etc. When in doubt, consider spaces at the left as levels of indentation, not countable spaces.

B. Runovers should always begin in cell 2, regardless of the current depth of indentation.

Example 7-1: Indented Material

```
results()    /* print table using array pointers */
{
    extern int *pA1,*pA2,*pA3;
    int i;
    printf("\n %10s%10s%10s%10s \n \n","i","A1[i]","A2[i]","A3[i]");
    for(i = 0;i <= MAX-1;i++)
        printf("%10d%10d%10d%10d \n",i,pA1[i],pA2[i],pA3[i]);
}
```

The image displays a large grid of 100 rows and 100 columns of dots. Each dot in the grid is replaced by a small cluster of dots, creating a complex, abstract pattern. The clusters vary in size and shape, ranging from small groups of 2 to 10 dots. The overall effect is a dense, textured field of dots, with some areas appearing more concentrated than others. The pattern is not uniform, showing a mix of small, isolated clusters and larger, more interconnected groups. The dots are arranged in a way that suggests a random or semi-random distribution, but the overall density and the presence of larger clusters give it a structured appearance.

Example 7-2: Indented Material

```
UnionOrderPackage =
{ &CompareAB →
  (lambda (x y)
    (cond ((&MemberA x) (cond ((&MemberA y) (&CompareA x y))
                              (t -1)))
    ((&MemberA y) 1)
    (t (&CompareB x y))))}
```

A large grid of 100 small 2x2 dot patterns arranged in a 10x10 grid. Each pattern is a 2x2 square of dots, with the dots in each square being either black or white, creating a binary image.

Note: The symbol for the right-pointing arrow was selected from the Nemeth Code and enclosed within the shape indicators.

VIII — Isolated Lower-Cell Signs

When a lower-cell sign is isolated (i.e., preceded and followed by a space or line end), dots 4-5-6 must immediately precede this sign. The lower-cell sign may be a single-digit number, quotation mark, apostrophe, comma, hyphen/minus sign, or semicolon. The resulting two-cell symbol signifies a single print character.

Example 8: Isolated Lower-Cell Signs

```
107 : CLEAR      YLEN 0 DO  XLEN 0 DO 0 I J UNIVERSE C! LOOP LOOP ;
```

IX — Countable Spaces

Countable spaces indicator: Dots 4-5-6, followed immediately by at least two full cells (dots 1-2-3-4-5-6, 1-2-3-4-5-6) :: :: :: :: ::

A. The countable spaces indicator is used when any series of five or more spaces within a print line is significant (e.g., when specific material which is surrounded by spaces must begin on a particular space within the line). These significant spaces, which are important to an accurate understanding of the material being transcribed, are called countable spaces. The first and last spaces of a countable space sequence are transcribed as empty cells (i.e., spaces), with a countable spaces indicator filling the remainder of the spaces to be counted. Each cell of a countable spaces sequence counts as one space, as does each of the spaces surrounding it.

Example 9-1: Countable Spaces Indicator

```
110 REM      SORT AN ARRAY INTO DESCENDING ORDER.
120
130 DIM X (100)
```

B. At least three full cells of the countable spaces indicator must appear on any runover line(s).

Example 9-2: Countable Spaces with Continuation Indicator

[illegible]

Example 9-3: Countable Spaces Beginning on Runover Line

```

6 : <A>      0 0 HERE 1+ CONVERT DROP DROP  ( extract variable # )
7           [COMPILE] LITERAL                ( make it a literal )
8           COMPILE <ADDR> ;                  ( dynamic part of address )
9 1 WIDTH !   ( set significant characters in name )
10 : @X      ( --- value ) ( fetch contents of variable x

```

X — Emphasis Indicators

Begin emphasis: Dots 4-5-6, 1-6 :: ::

End emphasis: Dots 4-5-6, 3-4 :: ::

A. Emphasis indicators enclose computer-related material which is emphasized for technical reasons (e.g., to distinguish what a student types from what the computer responds). Print methods of emphasizing this material may include underlining, italicizing, changing type fonts, changing color, etc.

Example 10-1: Emphasis Indicators

... for example, `chkdsk a:*.*/v` would check the diskette in drive A, check all files on it for fragmentation, and display the names of all files and directories.

A 4x100 grid of dots forming a complex pattern. The pattern consists of various geometric shapes and lines created by the arrangement of dots. The dots are arranged in a way that suggests a larger, more complex structure, possibly a face or a figure, though the details are abstract and difficult to discern. The pattern is composed of many small, repeating units that together form a larger, more complex whole.

Example 10-2: Emphasis Indicators

&Hdddd where *dddd* is a 1 to 4 digit sequence composed of hexadecimal ...

B. When more than one type style is used to emphasize technically significant material, a transcriber's option symbol should begin emphasis and a termination indicator should end emphasis in each case. A transcriber's note should explain the use of transcriber's option symbols. (See Section XIV for a discussion of transcriber's option symbols and Section XVI for a discussion of the termination indicator.)

Example 10-3: Emphasis Indicators with More Than One Type Style

```
enter <imp>
do
    imp O> 'restart' lab;
    'leave' lab O> imp;
exit <imp>
#)
```

Note: The symbol for the circle was selected from the Nemeth Code and enclosed within the shape indicators.

XI — Computer Braille Code Indicators

Begin Computer Braille Code indicator: Dots 4-5-6, 3-4-6 ⠠⠠

End Computer Braille Code indicator: Dots 4-5-6, 1-5-6 ⠨⠠

A. Computer Braille Code indicators need not enclose computer notation set off from the body of the text. Computer Braille Code should be entered when textual passages include obvious computer notation (e.g., material to be typed to the computer by the reader in a specific manner).

B. If the caps lock indicator, the begin emphasis indicator, the begin Nemeth code indicator, or the begin shape indicator begins embedded Computer Braille Code, the begin Computer Braille Code indicator need not be used; however, the end Computer Braille Code indicator must always be used to signal a return to regular text. The purpose of Computer Braille Code indicators is to clarify potentially ambiguous situations (e.g., the character preceding s in the filename item.s could be either a period or part of the “less” sign). Frequent switches between Computer Braille Code and the primary code being used should be avoided, particularly within the same braille line. Thus, unless a row or column of a table, a title, or other contiguous material can be transcribed entirely in the primary code, Computer Braille Code should be employed.

Example 11-1: Computer Braille Code Indicators

. . . the symbols of the form *1, *2, etc.

⠠⠠⠠⠠ ⠠⠠ ⠠⠠⠠⠠⠠⠠⠠⠠⠠ ⠠⠠ ⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠

Example 11-2: Avoiding Frequent Switches between Computer Braille Code and Primary Code

. . . the symbols of the form *1 and *2, etc.

⠠⠠⠠⠠ ⠠⠠ ⠠⠠⠠⠠⠠⠠⠠⠠⠠ ⠠⠠ ⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠ ⠠⠠⠠⠠⠠⠠ ⠠⠠⠠⠠

Example 11-3: Use of Computer Braille Code Indicators to Clarify Ambiguities

item.s

⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠

Example 12-2: Nemeth Code Indicators with Horizontal Fraction Line

... situation is clarified if one considers a different form of the equation of the ellipse:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

where a and b are the radii along the x and y ...

B. Nemeth code indicators should be written unspaced from the material they enclose. Within Nemeth code indicators, Nemeth code notation and rules should be followed. If a long passage includes several Nemeth code expressions, the begin Nemeth code indicator should precede the first Nemeth expression. The end Nemeth code indicator should be written after the last Nemeth expression in the passage has been transcribed. However, Computer Braille Code indicators should enclose computer notation within Nemeth code expressions. The end Nemeth code indicator returns the transcription to the baseline of writing.

Example 12-3: Nemeth Code Indicators within Computer Braille Code

Now we will cleverly initialize our added variable `d_norm` to 2β before entering the loop, so every pass we add 4β to `d_norm` before adding it, in turn, to the former value of `d`.

XIII — Shape Indicators

Begin shape indicator: Dots 4-5-6, 1-2-4-6 :: ::

End shape indicator: Dots 4-5-6, 1-5-6 :: ::

A. Shape indicators are used to enclose a brief series of symbols describing a shape not represented in the Computer Braille Code.

Example 13-1: Shape Indicators

$$S \sqsubseteq R = \llbracket \langle s, \text{add} \rangle \mid s \in S - R \rrbracket$$

Figure 1. The 100 most frequent words in the 1000-word corpus. The words are arranged in two rows, with the first row containing 50 words and the second row containing 50 words. The words are arranged in descending order of frequency, with the most frequent word at the top left and the least frequent word at the bottom right. The words are represented by a grid of dots, where each dot represents a letter. The words are arranged in two rows, with the first row containing 50 words and the second row containing 50 words. The words are arranged in descending order of frequency, with the most frequent word at the top left and the least frequent word at the bottom right. The words are represented by a grid of dots, where each dot represents a letter.

Note: The shape for the square with an interior minus was transcribed as an s followed by a minus and enclosed within the shape indicators. The left and right double-barred brackets were transcribed as lb and rb within the shape indicators.

B. When keys on a computer keyboard are depicted, the begin shape indicator should be followed by the letter k, the key legend, and the end shape indicator.

Example 13-2: Depicted Key

Hit **ENTER** key.

The image displays a sequence of 15 dot patterns, organized into three groups of five. Each group illustrates a progression of dots forming a shape, likely the letter 'E'. The first group shows a vertical column of 5 dots, followed by a horizontal row of 5 dots, and then a second vertical column of 5 dots. The second group shows a vertical column of 5 dots, followed by a horizontal row of 5 dots, and then a second vertical column of 5 dots. The third group shows a vertical column of 5 dots, followed by a horizontal row of 5 dots, and then a second vertical column of 5 dots.

C. Transcribers familiar with Nemeth code may select or devise symbols in accordance with Nemeth code conventions, thus enhancing consistency among braille codes.

Examples 13-3: Use of Nemeth Code within Shape Indicators

Characters existing only in the 'A' set:

& # % □ @

Equivalent characters from . . .

Note: The symbol for the square was selected from the Nemeth Code and enclosed within the shape indicators.

D. A transcriber's note at the beginning of the work or immediately prior to the first use of shape indicators should explain the symbol's meaning (if known), or its shape should be described or drawn.

XIV — Transcriber's Option Symbols

Primary transcriber's option symbol: Dots 4-5-6, 2-3-4-6 :: ::

Secondary transcriber's option symbol: Dots 4-5-6, 4-6 :: ::

A. When a character not represented by the Computer Braille Code occurs in a transcription with great frequency, the transcriber may use a transcriber's option symbol to represent that character. The minimum number of spaces required to produce the character is thus reduced from five to two. Transcriber's option symbols should be explained in a transcriber's note.

Example 14-1: Transcriber's Option Symbol Representing Non-ASCII Character (~~b~~)

```
RENAME %filename1[/ext1] [.psw] [:d] %TO%filename2[/ext2]
```

B. When the print text does not include a character used by the Computer Braille Code but does include a character not represented by the Computer Braille Code, the meaning of an unused Computer Braille Code character may be changed during that transcription (e.g., in an EBCDIC transcription, the unused Computer Braille Code tilde could represent the frequently used EBCDIC “not” sign). A transcriber’s note should explain such substitutions.

Example 14-2: Transcriber's Substitution (ASCII Tilde Becomes EBCDIC "Not" Sign)

EX = (P $\neg$ = NULL):

XV — Half-Line Shifts

Half-line shift up: Dots 4-5-6, 3-4-5-6 :: ::

Half-line shift down: Dots 4-5-6, 1-4-5-6 :: ::

End half-line shift: Dots 4-5-6, 1-5-6 :: ::

A. Half-line shifts up represent superscripts—and in some cases exponentiation—which typically appear within text. If exponentiation within text is indicated by the up arrow, the up arrow symbol (dots 4-5 ⋈) should be transcribed. Half-line shifts up terminate either with a termination indicator (see Section XVI) or at the next space, depending on applicable Computer Braille Code rules.

B. Half-line shifts down are used for transcribing subscripts and expressions in which the number base appears below normal line level. Half-line shifts down terminate either with a termination indicator (see Section XVI) or at the next space, depending on applicable Computer Braille Code rules.

Example 15-1: Half-Line Shifts

7.2.2 Data Initialization Statement. A data initialization statement is of the form:

DATA $k_1/d_1, k_2/d_2, \dots, k_n/d_n$

where:

- (1) Each k is a list containing names of variables and array elements
- (2) Each d is a list of constants and optionally signed constants.

The image displays a large grid of 1000 dots, arranged in a 10x100 pattern. Each dot is represented by a small black square. The dots are organized into a regular grid, with 10 rows and 100 columns. This visual representation corresponds to the number 1000 in Braille, where each dot represents a single Braille character.

Example 15-2: Half-Line Shift Terminated by a Space

. . . A listener may set NDAC low (shown between t_5 and t_6) but, it must do so before NRFD is set high at t_6 .

A 3x10 grid of 30 small clusters of black dots. Each cluster is composed of 10 dots arranged in various patterns, such as lines, squares, and irregular shapes. The patterns vary significantly from one cluster to the next, creating a diverse visual sequence.

C. Reference indicators designating footnotes are transcribed in the primary code being used, not as half-line shifts up.

Example 15-3: Reference Indicator (English Braille)

This chapter discusses the use of the Pascal* subsystem that operates under the environment . . .

***At least 160K bytes of main storage are required to install and use DOS Pascal.**

The image displays a large, intricate pattern of dots arranged in a grid-like structure. The dots are organized into rows and columns, forming a series of small, repeating units that resemble characters or symbols. The overall layout is dense and symmetrical, with the dots arranged in a way that suggests a larger, structured message or data set. The pattern is composed of many small, dark dots on a light background, creating a high-contrast visual effect.

Example 15-4: Reference Indicator (Nemeth Code)

```
10 PRINT
20 PRINT3
99 END
```

RUN⁴

4. If you are not using an interactive system, ignore the word **RUN** shown before each output.

**Computer Braille Code Symbols:
ASCII Order**

Note: Upper case letters may be transcribed as shown in the table, preceded by dots 4-5-6, or preceded by the caps lock indicator; applicable Computer Braille Code rules determine transcription method. "Reserved" designates currently unassigned symbols which are reserved for future use.

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
			space
!	⠗⠗	2-3-4-6	exclamation point
''	⠗⠗ -	5	quotation marks, double quotes
#	⠗⠗	3-4-5-6	number sign
\$	⠗⠗	1-2-4-6	dollar sign
%	⠗⠗	1-4-6	percent sign
&	⠗⠗	1-2-3-4-6	ampersand
'	⠗⠗	3	apostrophe, single quote
(	⠗⠗	1-2-3-5-6	left parenthesis
)	⠗⠗	2-3-4-5-6	right parenthesis
*	⠗⠗	1-6	asterisk, multiplication sign
+	⠗⠗	3-4-6	plus sign
,	⠗⠗	6	comma
—	⠗⠗	3-6	hyphen, minus sign
.	⠗⠗	4-6	period

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
/	⠇	3-4	slash, division sign
0	⠼	3-5-6	0
1	⠠	2	1
2	⠡	2-3	2
3	⠢	2-5	3
4	⠣	2-5-6	4
5	⠤	2-6	5
6	⠥	2-3-5	6
7	⠦	2-3-5-6	7
8	⠧	2-3-6	8
9	⠨	3-5	9
:	⠒	1-5-6	colon
;	⠒	5-6	semicolon
<	⠨	1-2-6	less than
=	⠒	1-2-3-4-5-6	equals
>	⠨	3-4-5	greater than
?	⠒	1-4-5-6	question mark
@	⠒	4	at

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
A	⠁	1	A
B	⠃	1-2	B
C	⠉	1-4	C
D	⠙	1-4-5	D
E	⠑	1-5	E
F	⠋	1-2-4	F
G	⠎	1-2-4-5	G
H	⠓	1-2-5	H
I	⠊	2-4	I
J	⠗	2-4-5	J
K	⠅	1-3	K
L	⠕	1-2-3	L
M	⠍	1-3-4	M
N	⠝	1-3-4-5	N
O	⠥	1-3-5	O
P	⠏	1-2-3-4	P
Q	⠑	1-2-3-4-5	Q
R	⠞	1-2-3-5	R

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
S	⠠	2-3-4	S
T	⠡	2-3-4-5	T
U	⠢	1-3-6	U
V	⠣	1-2-3-6	V
W	⠤	2-4-5-6	W
X	⠥	1-3-4-6	X
Y	⠦	1-3-4-5-6	Y
Z	⠧	1-3-5-6	Z
[	⠨	2-4-6	left bracket
\	⠩	1-2-5-6	backslash
]	⠪	1-2-4-5-6	right bracket
↑	⠫	4-5	up arrow/caret
—	⠬ ⠬	4-5-6, 4-5-6	underscore
`	⠭ ⠬	4-5-6, 4	grave accent
a	⠼	1	a
b	⠽	1-2	b
c	⠿	1-4	c
d	⠻	1-4-5	d

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
e	⠠	1-5	e
f	⠠	1-2-4	f
g	⠠	1-2-4-5	g
h	⠠	1-2-5	h
i	⠠	2-4	i
j	⠠	2-4-5	j
k	⠠	1-3	k
l	⠠	1-2-3	l
m	⠠	1-3-4	m
n	⠠	1-3-4-5	n
o	⠠	1-3-5	o
p	⠠	1-2-3-4	p
q	⠠	1-2-3-4-5	q
r	⠠	1-2-3-5	r
s	⠠	2-3-4	s
t	⠠	2-3-4-5	t
u	⠠	1-3-6	u

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
v	⠠	1-2-3-6	v
w	⠡	2-4-5-6	w
x	⠢	1-3-4-6	x
y	⠣	1-3-4-5-6	y
z	⠤	1-3-5-6	z
{	⠠⠠	4-5-6, 2-4-6	left brace
	⠠⠡	4-5-6, 1-2-5-6	vertical bar
}	⠠⠢	4-5-6, 1-2-4-5-6	right brace
~	⠠⠣	4-5-6, 4-5	tilde

Additional Computer Braille Code Symbols

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
	⠠⠠	4-5-6, 2-3-4-6	transcriber's option symbol
	⠠⠡	4-5-6, 4-6	transcriber's option symbol
	⠠⠢	4-5-6, 3-4-5-6	half-line shift up
	⠠⠣	4-5-6, 1-4-5-6	half-line shift down
	⠠⠤	4-5-6, 1-5-6	end half-line shift

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
	⠠⠠	4-5-6, 1-2-4-6	begin shape indicator
	⠡⠠	4-5-6, 1-5-6	end shape indicator
	⠠⠠⠠	4-5-6, 3-4-6	begin Computer Braille Code
	⠡⠠⠠	4-5-6, 1-5-6	end Computer Braille Code
	⠠⠠	4-5-6, 1-4-6	begin Nemeth code
	⠡⠠	4-5-6, 1-5-6	end Nemeth code
	⠠⠠	4-5-6, 1-2-3-4-6	continuation indicator
	⠠⠠⠠	4-5-6, 1-2-3-5-6	reserved
	⠠⠠⠠	4-5-6, 2-3-4-5-6	reserved
	⠠⠠⠠	4-5-6, 1-2-3-4-5-6	reserved
	⠠⠠	4-5-6, 1-6	begin emphasis
	⠡⠠	4-5-6, 3-4	end emphasis
	⠠	4-5-6	shift indicator, isolated lower-cell indicator
	⠠⠠⠠	4-5-6, 3-4-5	caps lock indicator
	⠠⠠⠠	4-5-6, 1-2-6	caps release indicator
	⠠⠠⠠⠠⠠⠠	(space)4-5-6, 1-2-3-4-5-6, 1-2-3-4-5-6, . . . (space)	countable spaces indicator

Computer Braille Code Symbols:

Braille Order

Note: Upper case letters may be transcribed as shown in the table or preceded by the caps lock indicator; applicable Computer Braille Code rules determine transcription method. "Reserved" designates currently unassigned symbols which are reserved for future use.

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
			space
A or a	⠁	1	A or a
B or b	⠃	1-2	B or b
C or c	⠉	1-4	C or c
D or d	⠙	1-4-5	D or d
E or e	⠑	1-5	E or e
F or f	⠥	1-2-4	F or f
G or g	⠭	1-2-4-5	G or g
H or h	⠓	1-2-5	H or h
I or i	⠏	2-4	I or i
J or j	⠛	2-4-5	J or j
K or k	⠕	1-3	K or k
L or l	⠕	1-2-3	L or l
M or m	⠍	1-3-4	M or m
N or n	⠝	1-3-4-5	N or n

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
O or o	⠠	1-3-5	O or o
P or p	⠡	1-2-3-4	P or p
Q or q	⠢	1-2-3-4-5	Q or q
R or r	⠣	1-2-3-5	R or r
S or s	⠤	2-3-4	S or s
T or t	⠥	2-3-4-5	T or t
U or u	⠦	1-3-6	U or u
V or v	⠧	1-2-3-6	V or v
W or w	⠨	2-4-5-6	W or w
X or x	⠩	1-3-4-6	X or x
Y or y	⠪	1-3-4-5-6	Y or y
Z or z	⠫	1-3-5-6	Z or z
&	⠠	1-2-3-4-6	ampersand
=	⠠	1-2-3-4-5-6	equals
(	⠠	1-2-3-5-6	left parenthesis
!	⠠	2-3-4-6	exclamation point
)	⠠	2-3-4-5-6	right parenthesis
*	⠠	1-6	asterisk, multiplication sign

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
<	⠨	1-2-6	less than
%	⠨	1-4-6	percent sign
?	⠨	1-4-5-6	question mark
:	⠨	1-5-6	colon
\$	⠨	1-2-4-6	dollar sign
]	⠨	1-2-4-5-6	right bracket
\	⠨	1-2-5-6	backslash
[	⠨	2-4-6	left bracket
1	⠨	2	1
2	⠨	2-3	2
3	⠨	2-5	3
4	⠨	2-5-6	4
5	⠨	2-6	5
6	⠨	2-3-5	6
7	⠨	2-3-5-6	7
8	⠨	2-3-6	8
9	⠨	3-5	9
0	⠨	3-5-6	0

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
/	⠇	3-4	slash, division sign
+	⠇	3-4-6	plus sign
#	⠇	3-4-5-6	number sign
>	⠇	3-4-5	greater than
'	⠇	3	apostrophe, single quote
—	⠇	3-6	hyphen, minus sign
@	⠇	4	at
↑	⠇	4-5	up arrow/caret
	⠇	4-5-6	shift indicator, isolated lower-cell indicator
A	⠇⠇	4-5-6, 1	A
B	⠇⠇	4-5-6, 1-2	B
C	⠇⠇	4-5-6, 1-4	C
D	⠇⠇	4-5-6, 1-4-5	D
E	⠇⠇	4-5-6, 1-5	E
F	⠇⠇	4-5-6, 1-2-4	F
G	⠇⠇	4-5-6, 1-2-4-5	G
H	⠇⠇	4-5-6, 1-2-5	H
I	⠇⠇	4-5-6, 2-4	I

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
J	⠠⠵	4-5-6, 2-4-5	J
K	⠠⠴	4-5-6, 1-3	K
L	⠠⠢	4-5-6, 1-2-3	L
M	⠠⠠	4-5-6, 1-3-4	M
N	⠠⠠⠠	4-5-6, 1-3-4-5	N
O	⠠⠠⠠	4-5-6, 1-3-5	O
P	⠠⠠⠠	4-5-6, 1-2-3-4	P
Q	⠠⠠⠠	4-5-6, 1-2-3-4-5	Q
R	⠠⠠⠠	4-5-6, 1-2-3-5	R
S	⠠⠠⠠	4-5-6, 2-3-4	S
T	⠠⠠⠠	4-5-6, 2-3-4-5	T
U	⠠⠠⠠	4-5-6, 1-3-6	U
V	⠠⠠⠠	4-5-6, 1-2-3-6	V
W	⠠⠠⠠	4-5-6, 2-4-5-6	W
X	⠠⠠⠠	4-5-6, 1-3-4-6	X
Y	⠠⠠⠠	4-5-6, 1-3-4-5-6	Y
Z	⠠⠠⠠	4-5-6, 1-3-5-6	Z

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
	⠠⠠	4-5-6, 1-2-3-4-6	continuation indicator
	⠠⠠	4-5-6, 1-2-3-4-5-6	reserved
	⠠⠠⠠⠠⠠⠠	(space)4-5-6, 1-2-3-4-5-6, 1-2-3-4-5-6, . . .(space)	countable spaces indicator
	⠠⠠	4-5-6, 1-2-3-5-6	reserved
	⠠⠠	4-5-6, 2-3-4-6	transcriber's option symbol
	⠠⠠	4-5-6, 2-3-4-5-6	reserved
	⠠⠠	4-5-6, 1-6	begin emphasis
	⠠⠠	4-5-6, 1-2-6	caps release indicator
	⠠⠠	4-5-6, 1-4-6	begin Nemeth code
	⠠⠠	4-5-6, 1-4-5-6	half-line shift down
	⠠⠠	4-5-6, 1-5-6	termination indicator
	⠠⠠	4-5-6, 1-5-6	end Computer Braille Code
	⠠⠠	4-5-6, 1-5-6	end Nemeth code
	⠠⠠	4-5-6, 1-5-6	end shape indicator
	⠠⠠	4-5-6, 1-5-6	end half-line shift
	⠠⠠	4-5-6, 1-2-4-6	begin shape indicator

<u>Print</u>	<u>Braille</u>	<u>Dots</u>	<u>Meaning</u>
}	⠼ ⠼	4-5-6, 1-2-4-5-6	right brace
	⠼ ⠼	4-5-6, 1-2-5-6	vertical bar
{	⠼ ⠼	4-5-6, 2-4-6	left brace
	⠼ ⠼	4-5-6, 3-4	end emphasis
	⠼ ⠼	4-5-6, 3-4-6	begin Computer Braille Code
	⠼ ⠼	4-5-6, 3-4-5-6	half-line shift up
	⠼ ⠼	4-5-6, 3-4-5	caps lock indicator
`	⠼ ⠼	4-5-6, 4	grave accent
~	⠼ ⠼	4-5-6, 4-5	tilde
—	⠼ ⠼	4-5-6, 4-5-6	underscore
	⠼ ⠼	4-5-6, 4-6	transcriber's option symbol
.	⠼	4-6	period
''	⠼	5	quotation marks, double quotes
;	⠼	5-6	semicolon
,	⠼	6	comma

